HP OpenView

Storage Mirroring application notes

Backup Enhancement

Part number: T2558-88079
Second edition: June 2006

Legal and notice information

Storage Mirroring Backup Enhancement application notes

Document overview

This document is a Storage Mirroring application note. An application note provides guidelines on the use of Storage Mirroring in a specific environment.

This document contains:

- **Document Overview**—Explains what an application note contains, how it should be used, what you need to know before trying to use the application note, and where you can go for more information.
- **Solution Overview**—Explains how the application works with Storage Mirroring and describes the considerations that you must weigh when implementing your Storage Mirroring solution. Review this section to make sure that you understand the theory involved with using Storage Mirroring and your application. Includes both basics, such as system requirements, as well as configuration and environment-specific topics, such as interactions with specific clients or special considerations for WAN (Wide Area Network) environments. Pay special attention to those topics that are directly related to your environment.
- **Sample Implementations**—Describes specific examples of how to use Storage Mirroring for this solution. This includes information about the specific system setup used in the sample implementation. Use these procedures as a guideline for creating your own implementation. Because no two environments or configurations are exactly the same, you will probably need to implement additional or different steps than what is documented here in order to make the solution work in your environment.About this document

Intended audience

This document is written for network and application administrators who have a working understanding of the applications and environments where the Storage Mirroring solution is to be deployed. You many need to expand on the documented information in order to customize the solution to fit your environment.

Expectations

Application notes are intended to provide a framework for configuring a Storage Mirroring solution in a specific environment and to draw attention to decisions you will need to make when configuring your solution.

Because there are an infinite number of possible configuration, network, and environment scenarios, application notes contain general configuration guidelines as well as an example configuration procedure that has been tested for a specific environment.

This document assumes that you are comfortable working with your operating system and Storage Mirroring.

Related documentation

Before you begin to configure your solution, make sure that you have complete documentation for your operating system, application, and Storage Mirroring. This application note does not provide step-by-step instructions for using standard operating system, application, and Storage Mirroring functionality.

The following documents contain additional information that you may need while setting up this solution:

Storage Mirroring user's guide or online documentation

Reference guides or documentation for your storage solution

Getting help

Hewlett-Packard has application notes that describe how to configure Storage Mirroring with a variety of popular third-party applications. These application notes are available on the Storage Mirroring web site: http://h18006.www1.hp.com/products/storage/software/sm/index.html.

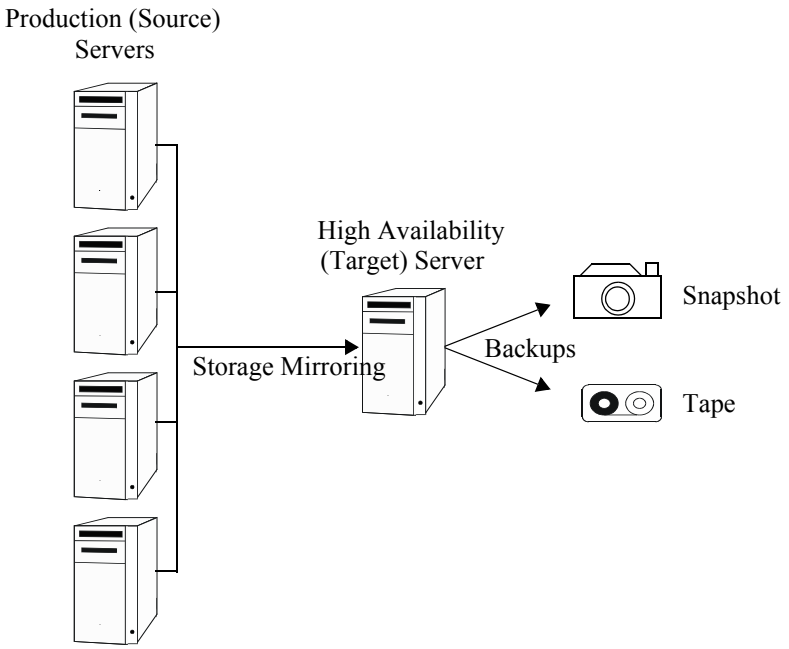
For help using Storage Mirroring, refer to the Storage Mirroring online manual or online help.

Solution overview

The rapid growth in storage brought on by the Internet and distributed computing has placed nearly impossible demands on administrators responsible for protecting corporate data assets. The backup window has shrunk to nearly zero and tape backup systems can introduce significant overhead to a production server, seriously impacting its performance. While the importance of backups increases, the impact of periodic full system backups is obvious. Even nightly incremental backups dominate processing while they examine every file system object and then read all files that have changed in their entirety for backup. Performing this process across a network adds additional overhead as the entire process happens across the wire.

These days permanent point in time storage and recovery, like that provided by periodic tape backup and point in time snapshot technology, is required. And despite the fact that Storage Mirroring cannot provide a way to retrieve historical file versions or files that may have been previously deleted by users, Storage Mirroring can enhance tape backup systems and snapshot technology because it is designed to capture changes and apply those changes to the target as quickly as possible to keep the target synchronized.

Storage Mirroring can enhance the backup process by continuously replicating critical data to centralized servers and using tape backup systems and snapshot technology to backup the replica rather than the production servers. Using Storage Mirroring offloads the burden of periodic tape backups from multiple production servers to a dedicated backup server and makes centralized tape backup a reality, significantly reducing management cost and improving reliability. Regardless of a file's state on the source, on the target every file is closed and available for consistent backup at any point in time.



Sample Backup Schedule

Sun	Mon	Tue	Wed	Thur	Fri	Sat
	1 S	2 S	3 S	4 S	5 S	6 WT
7	8 S	9 S	10 S	11 S	12 S	13 WT
14	15 S	16 S	17 S	18 S	19 S	20 WT
21	22 S	23 S	24 S	25 S	26 S	27 WT
28	29 S	30 S	31 MT			

S = Daily Snapshot Backup

WT = Weekly Tape Backup

MT = Monthly Tape Backup

In addition to enhancing the backup process, Storage Mirroring can take the process a step further and eliminate the often lengthy time required to restore from a tape backup by removing the tape restore process all together. Storage Mirroring failover capabilities allow a target machine to stand in for a source in the event of a failure, using the already online disk-based replica and removing the need to restore from tape.

Storage Mirroring software's flexibility allows many different enhancements to suit your unique environment and needs. This document covers various backup enhancements and alternatives which can be used with Storage Mirroring version 4.3.x. If you are using an earlier version of Storage Mirroring, contact technical support for additional information. This document is intended for network administrators with experience installing, configuring, and maintaining network applications including Storage Mirroring.

Identifying a backup method

Because of Storage Mirroring's patented STAR technology, Sequential Transfer Asynchronous Replication, Storage Mirroring replication follows the same write sequence within and across multiple files, providing complete data integrity at all times. Because, at any given moment, the target represents a single point in time from the source, the target is crash consistent. But for some applications, crash consistency may not be adequate. Your backup may require that the source data be in a quiescent, or latent, state, similar to an application checkpoint. This too can be accomplished with Storage Mirroring. With the flexibility of Storage Mirroring, you can choose the backup method that best suits your environment.

Backup Approach	Storage Mirroring Interaction	Advantages	Disadvantages
Backup or snapshot of the source data	None – The backup is done as usual while Storage Mirroring provides continuous replication of data to target. [1]	No changes to the existing backup are required	Backup impacts the performance of the production server Long recovery time from tape [2]
Scheduled backup or snapshot of the Storage Mirroring target data	Storage Mirroring automatically delays writing changes to files if they are in the process of being backed up. Storage Mirroring may also be paused during backup to provide a time-frozen state during the entire backup.	• Offload backup and snapshot management to the target server • Backups can be performed at any time or multiple times a day	• Backups do not represent specific source times or states • Represents crash consistent state [3]
Event driven backup or snapshot of the Storage Mirroring target data	Insert a task command in the Storage Mirroring replication queue based on a source application event or checkpoint. This task command can trigger a backup or snapshot on the target when the target file system reaches that point in time from the source file system state.	Same advantages as a scheduled backup or snapshot of the Storage Mirroring target data plus: • Backups represent a specific source time and file state(s) • Backups can be triggered from the source at any time and queued for remote processing	• Requires more complicated scripting • May require temporary disruption to application to quiesce files

1. While significant benefits can be achieved by performing backup and snapshots of target data replicated by Storage Mirroring, it is not necessary to change your existing source backup strategy if you choose not to. It is possible for Storage Mirroring to provide continuous replication of the same files being backed up to tape or included on a snapshot of a volume. It may be desirable to continue to backup the Operating System System State and other files that are not replicated by Storage Mirroring from the source or perform archival tasks such as month end backups directly from the source if desired.
2. By also adding in Storage Mirroring failover and restoration capabilities, you can reduce the time needed to restore a failed source. In the event of a failure, the target stands in for the failed source and users access the replica of the data and applications from the target. When the source is ready to be brought back online, a Storage Mirroring restoration synchronizes the source with the data changes that occurred on the target during the down-time. The only downtime the users will have is during the failback process. For detailed information, see the Storage Mirroring *uer's guide*.
3. A crash consistent state means files represent the same point in time and would be as if the source machine was powered off or crashed abruptly during operation. All applications need to be able to recover from files left in this state, however, it may take longer for the application to recover from this state. Therefore, for some applications it may be desirable to perform backups of a known quiesced file state such as when an application is temporarily shut down. Use event driven backup or snapshot to accomplish this.

Backup notes

Review the general backup related notes below.

- **Incremental backups**—If you are performing incremental or differential backups on your target machine, you need to make sure that your backup software is using an appropriate flag to identify what files have been updated since the last backup. For example, Storage Mirroring does not replicate

the last access time if it is the only thing that has changed. The suggested method for incremental or differential backups is to use the last modified date on the file versus the date of the last backup.

- **Registry Settings and System Files**—Storage Mirroring should not be used to replicate registry settings or system files, although Storage Mirroring can be used to replicate file-based backups of this information. For example, you could use NTBackup on Windows® 2000 to create a file containing a duplicate copy of the System State, which includes the registry. Storage Mirroring can then replicate this file to the target. On Windows NT® 4.0, you can use the Regback tool from the Windows NT Resource Kit. This tool also creates a file, containing the registry information, which can be replicated to the target. See your Windows documentation for complete details on using these commands and tools.
- **Backing Up Deleted Files**—A feature of Storage Mirroring, called ignore delete operations, allows you to keep files on the target machine even if they are deleted on the source. When a file is deleted on the source, that delete operation is ignored on the target. (All edits to files on the source are still replicated to the target; only deletions of whole files are ignored.) By default, this option is disabled. Enabling it would give you the opportunity to make backups of these files in the event they are needed in the future, after they have been deleted from the source. For instructions on enabling this option, see the Storage Mirroring *user's guide*.

NOTE: If a file is deleted using the Windows Explorer or My Computer, from the file system perspective, the file is not actually deleted but is moved to the Windows Recycle Bin. Because Storage Mirroring sees this as a move to outside of the replication set and not as a delete, the file will still be deleted from the target even if you have ignore delete operations enabled.

If delete operations are ignored long enough, the potential exists for the target to run out of space. In that case, you can manually delete files from the target to free space for further replication or use the automatic move or delete capabilities of Storage Mirroring orphan files feature. For complete details on orphans files, see the *Storage Mirroring user's guide*.

- **Applications with Interdependent Files**—Backups occur sequentially from the first file to the last file. Therefore, when you are using applications that have interdependent files, such as a database application whose database and log files must be synchronized, Storage Mirroring cannot be actively updating files on the target while the backup is running or there becomes an opportunity for interdependent files to become mismatched, causing a corrupt application on the backup. For example, suppose the following scenario occurs on a target machine that contains a replica of a database:

1. The backup process which is currently running sequentially through the files, reaches the database log file and starts writing the file to tape. At the same time, Storage Mirroring receives additional updates to the database. The database file is updated but since the log file is in use by the backup, the log operation is placed on the Storage Mirroring queue on the target.
2. When the log file is finished being backed up, the backup process continues with the next file, which is not necessarily the database that corresponds with that log file.
3. Since the log file is no longer in use, Storage Mirroring applies the log operation from the Storage Mirroring queue.
4. Eventually, the backup process reaches the database file and writes it to tape.

At this point, the database file on the tape backup contains an extra update that the log file does not. The two files do not correspond and the database on the tape backup will not be time consistent.

Any of the backup enhancements in this document can be used in conjunction with applications with interdependent files thus preventing the target from committing data to disk during the backup.

Modifying the sample script files

After you modify the sample scripts, save them with a new name to remove the `SAMPLE_` prefix. Copy the scripts to the directory where Storage Mirroring is installed.

The sample batch files provided are only examples. Because no two environments or configurations are exactly the same, you **MUST** modify the sample scripts in order to make the solution work in your environment.

Configuring memory usage

Storage Mirroring uses memory to queue operations and data on both the source and target. Since the source server is typically running a production application, it is important that the amount of memory Storage Mirroring and the other applications use does not exceed the amount of RAM in the system. If the applications require more memory than there is RAM, the system will begin to swap pages of memory to disk and the system performance will degrade.

For instance, SQL Server will use all of the available system memory when needed by default, and it may use almost all of the system memory during high-load operations. These high-load operations are precisely what cause Storage Mirroring to need memory to queue the data being changed by SQL Server. On a server with 1 GB of RAM running SQL Server and Storage Mirroring, you might configure SQL Server to use only 512 MB and Storage Mirroring to use 256 MB, leaving 256 MB for the operating system and other applications on the system. Many other server applications will use almost all system memory by default, so it is important to check and configure applications appropriately, particularly on high-capacity servers.

Sample Implementation

This section describes an example of how to configure Storage Mirroring for backup enhancement. Use these procedures as a guideline for creating your own implementation. Because no two environments or configurations are exactly the same, you will probably need to implement additional or different steps than what is documented here in order to make the solution work in your environment.

The remainder of this document includes additional information and script files, if applicable, for several backup methods. Keep in mind, these are not the only methods available.

Pausing Storage Mirroring

Storage Mirroring allows you to pause execution of operations on the target. New operations are queued until execution on the target is resumed, at which time the queued operations are processed. This allows you to pause execution on the target, run a backup, and then resume execution on the target.

Pausing and resuming the target can be manually initiated through the Management Console, Text Client or you can use DTCL commands to script it. For complete details on using the Management Console or Text Client, see the *Storage Mirroring user's guide*. The following instructions contain step-by-step procedures for incorporating the DTCL commands used in the Text Client into a DTCL script which will allow you to automate the pause and resume process from a batch file.

1. Prior to the backup process starting, you will want to pause execution of operations on the target. To do this, create a batch file called `prebackup.txt` using the sample file below.

NOTE: After you modify the sample scripts, save them with a new name to remove the sample_ prefix. Copy the scripts to the directory where Storage Mirroring is installed.

The sample batch files provided are only examples. Because no two environments or configurations are exactly the same, you **MUST** modify the sample scripts in order to make the solution work in your environment.

SAMPLE_PREBACKUP.TXT

```
rem ***Sample*** script to pause execution of operations on the target prior to starting a backup

rem This sample batch file is provided as an example only. Because no two
rem environments or configurations are exactly the same, you MUST modify
rem this script in order to make the solution work in your environment.

rem Substitute the name of your target machine, username, password, and domain in the variable
rem definitions below.

$TheTarget = "name";
$User = "username";
$Pass = "password";
$TheDomain = "domain";

login $TheTarget $User $Pass $TheDomain;
target $TheTarget;
pausetarget $TheTarget;
```

NOTE: Because the following files use the login command, which displays the password of the user ID specified, you may not want to use the network administrator account. You can create a user account specifically for this purpose, add it to the Storage Mirroring Admin security group, and grant it the minimal access necessary to complete this task.

2. Create a batch file to run this script using the sample batch file below.

SAMPLE_PREBACKUP.BAT

```
rem ***Sample*** batch file to run the prebackup.txt script

rem This sample batch file is provided as an example only. Because no two
rem environments or configurations are exactly the same, you MUST modify
rem this script in order to make the solution work in your environment.

cd c:\program files\StorageMirroring
cmd /c DTCL -f "c:\program files\StorageMirroring\prebackup.txt"
```

3. After the backup process is complete, you will want to resume execution of operations on the target. To do this, create a batch file called `postbackup.txt` using the sample file below. Save this file to the location where Storage Mirroring is installed.

SAMPLE_POSTBACKUP.TXT

```
rem ***Sample*** script to resume execution of operations on the target after the backup is complete

rem This sample batch file is provided as an example only. Because no two
rem environments or configurations are exactly the same, you MUST modify
rem this script in order to make the solution work in your environment.

rem Substitute the name of your target machine, username, password, and domain in the variable
rem definitions below.

$TheTarget = "name";
$User = "username";
$Pass = "password";
$TheDomain = "domain";

login $TheTarget $User $Pass $TheDomain;
target $TheTarget;
resumetarget $TheTarget;
```

4. Create a batch file to run this script using the sample batch file below.

SAMPLE_POSTBACKUP.BAT

```
rem ***Sample*** batch file to run the postbackup.txt script

rem This sample batch file is provided as an example only. Because no two
rem environments or configurations are exactly the same, you MUST modify
rem this script in order to make the solution work in your environment.

cd c:\program files\StorageMirroring
cmd /c DTCL -f "c:\program files\StorageMirroring\postbackup.txt"
```

5. Run `prebackup.bat` before starting your backup and run `postbackup.bat` after the backup is complete.

You can also incorporate these two scripts into an automated script that runs through your backup software.

NOTE: Depending on the length of time required to complete your backup, Storage Mirroring may not be able to queue all of the replication data. If the queue is filled, Storage Mirroring will automatically disconnect the connections and attempt to reconnect them. This is called an auto-disconnect. If you are experiencing frequent auto-disconnects because the queues are filled while the backup is processing, you can:

- Increase the amount of disk space on the volume where the Storage Mirroring queue is located or move the extended queue to a larger volume
- Disable auto-reconnect and reconnect manually or in a post-backup DTCL script (It may also be desirable to script a DTCL disconnect command in a pre-backup script and reconnect in the post-backup script.)
- Create a DTCL script to disconnect Storage Mirroring before the backup and reconnect and remirror after the backup is complete

See the *Storage Mirroring user's guide* for additional details.

Inserting task commands during replication

Storage Mirroring allows you to insert and run tasks at various points during the replication of data. Because the tasks are user-defined, you can achieve a wide variety of goals with this feature. For example, you might insert a task to create a snapshot or backup on the target after a certain segment of data from the source has been applied on the target. This allows you to coordinate a point-in-time backup with real-time replication.

In order to quantify the certain segment of data from the source, you need to be able to identify when the application is stable, which is usually when all of the data has been written to disk. This can be triggered by stopping the service. With task command processing, you can stop the source service just long enough to identify that stopped point in time as a stable state, insert a task at that point into the Storage Mirroring replication queue to trigger a backup or snapshot on the target, and then restart the service. Here is how the process would work.

1. Storage Mirroring and an application are both running on the source. Only Storage Mirroring is running on the target.
2. The application data is changing on the source and Storage Mirroring is capturing those data changes and transmitting them to the target.
3. A script is launched (either manually or perhaps scheduled by the Windows NT Scheduler) that stops the application service on the source, pauses to give the service time to shutdown and write the data to disk, initiates a Storage Mirroring task command, and then restarts the application service on the source.
4. The Storage Mirroring task command is transmitted, inline with the source replication data, to the target.
5. The data is applied on the target as it is received. Since the task command was inserted inline, the replication data from the source is applied to the target first. When the target gets to the Storage Mirroring task command, the target data will be in the exact same state as the source data when the

source application service was stopped. Since this was a stable point on the source, it is also a stable point on the target.

6. The target processes the Storage Mirroring task command and completes whatever task is defined, perhaps a snapshot or backup. Since the Storage Mirroring task command is user-defined, you can insert any valid executable or batch file.

NOTE: The following batch file should be stored on the source and run manually when desired or can be scheduled using the Windows NT Scheduler. See your Microsoft Windows reference guide for details on scheduling.

For complete details on the DTCL queuetask command that is used, see the Storage Mirroring user's guide. For complete details on Microsoft commands, see your Microsoft reference guide.

SAMPLE_SQL_BACKUP.BAT

```
REM ***Sample*** batch file that stops the Microsoft SQL Server 2000 services on the
REM source, pauses to allow the source to write all of the SQL data to the source, inserts a Storage
Mirroring
REM task command into the Storage Mirroring replication process, and then restarts the SQL services.

REM This sample batch file is provided as an example only. Because no two
REM environments or configurations are exactly the same, you MUST modify
REM this script in order to make the solution work in your environment.

REM Storage Mirroring task command processing must be enabled and there must be an active Storage Mirroring
REM connection for this process to function properly. See the Storage Mirroring User's Guide for assistance
REM in enabling task command processing and establishing a connection.

REM The following line calls a batch file which stops the Microsoft SQL Server 2000 services on
REM the source. This batch file should be stored on the source.

c:\scripts\StopServices.bat

REM The following line pauses the execution of this batch file for 120 seconds (2 minutes) so that any
REM remaining application data can be written to disk on the source. This command is available from the
REM Windows 2000/NT Resource Kit. If you do not have the Resource Kit, you will need to determine
REM another method to delay script processing. You may need to adjust the setting to accomodate the
REM amount of data your application is processing and the speed of your environment.

sleep 120

REM Since the source service is now stopped on the source and all of the data has been written to disk,
REM the application is now in a stable state. When the target reaches this exact point, you want to
REM initiate the backup. The following line calls a batch file which inserts a Storage Mirroring task
command to
REM initiate NTBackup on the target. The batch file should be stored on the source.

c:\scripts\DoBackup.bat

REM Now that the command to perform the NTBackup been inserted, inline with the data, the service can be
REM restarted. New updated data will fall inline behind the task command that was just inserted.

REM The following line calls a batch file which starts the Microsoft SQL Server 2000 services on
REM the source. This batch file should be stored on the source.

c:\scripts\StartServices.bat
```

The batch files called in this process are provided on the following pages.

SAMPLE_STOPSERVICES.BAT file used in SQL_BACKUP.BAT

```
REM ***Sample*** batch file that stops the Microsoft SQL Server 2000 services without
REM requiring administrator interaction.

rem This sample batch file is provided as an example only. Because no two
rem environments or configurations are exactly the same, you MUST modify
rem this script in order to make the solution work in your environment.


net stop "Distributed Transaction Coordinator"
net stop "Message Queuing"
net stop "MSSQLServer" /y
net stop "SQLServerAgent"
```

SAMPLE_DOBACKUP.BAT file used in SQL_BACKUP.BAT

```
REM ***Sample*** batch file that runs the Storage Mirroring Command Line Client File Entry.

rem This sample batch file is provided as an example only. Because no two
rem environments or configurations are exactly the same, you MUST modify
rem this script in order to make the solution work in your environment.

cd c:\Program Files\DoubleTake
cmd /c DTCL -f "c:\Program Files\DoubleTake\task.txt"
```

SAMPLE_TASK.TXT file used in DOBACKUP.BAT

```
REM ***Sample*** DTCL file that logins into a source and target server and inserts task
REM commands that will trigger a backup on the target.

REM This sample batch file is provided as an example only. Because no two
REM environments or configurations are exactly the same, you MUST modify
REM this script in order to make the solution work in your environment.

REM Substitute the name of your source and target machines as well as the login credentials. If you do not
REM want the login credentials of the administrator account exposed in this file, you can use another
REM account, as long as it is a member of the Storage MirroringStorage Mirroring Admin security group on
both the source and
REM target servers.

$TheSource = "sourcename";
$TheTarget = "targetname";
$TheUserName = "username";
$ThePassword = "password";
$TheDomain = "domain.com";
login $TheSource $TheUserName $ThePassword $TheDomainName;
login $TheTarget $TheUserName $ThePassword $TheDomainName;
queuetask backup_process to $TheTarget onexecute=BackupCommand.bat timeout=forever;

REM Because there are multiple arguments used in the NTBackup command, they would need to be enclosed in
REM quotation marks for Storage Mirroring to process them properly. But because the arguments required for
the
REM NTBackup command already use quotation marks, nested quotation marks would not be processed properly.
REM Therefore, the task is contained in its own batch file.
```

SAMPLE_BACKUPCOMMAND.BAT file used in TASK.TXT

```
REM ***Sample*** batch file to perform a backup. The command used is for NTBackup. The
REM command used will perform a copy backup of the local C: drive. The backup will be named "Backup C"
REM and the backed up files and folders will be appended to the tape named "Backup 1." All other options
REM will default to those specified in the backup program.

REM This sample batch file is provided as an example only. Because no two
REM environments or configurations are exactly the same, you MUST modify
REM this script in order to make the solution work in your environment.

ntbackup backup c:\ /j "Backup C" /a /t "Backup 1" /m copy

REM Substitute any command or series of commands for your particular backup strategy. For example, you
REM use VSSADMIN CREATE SHADOW /FOR=F:if you were using Windows 2003 Volume Shadow copy.
```

SAMPLE_STARTSERVICES.BAT file used in SQL_BACKUP.BAT

```
REM ***Sample*** batch file that starts the Microsoft SQL Server 2000 services.

REM This sample batch file is provided as an example only. Because no two
REM environments or configurations are exactly the same, you MUST modify
REM this script in order to make the solution work in your environment.

net start "Distributed Transaction Coordinator"
net start "Message Queuing"
net start "MSSQLServer"
net start "SQLServerAgent"
```

Rotating replicas

Storage Mirroring itself can also be used to create multiple copies of your data on the target including copies that are not actively being updated with new changes. This can be achieved by using Storage Mirroring to connect the same replication set to multiple targets or target locations. This would grant you the time and availability of idle files on the inactive replica to perform a backup.

For example, you might have two replicas in which one is active from midnight to noon and the other is active from noon to midnight. You can start and stop these replicas using Storage Mirroring DTCL commands to script an automated process. The two files below are an example of two DTCL scripts which could be used to replicate data to two different locations on the same target.

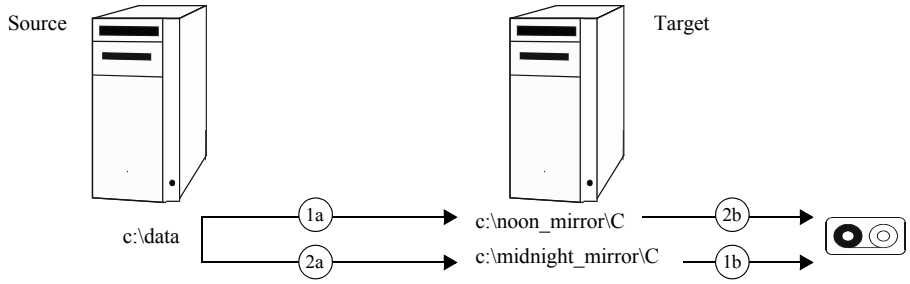
SAMPLE_Noon.dtcl

```
# ***Sample*** script to be run at noon that stops one connection and starts a second connection.#

# This sample batch file is provided as an example only. Because no two#
# environments or configurations are exactly the same, you MUST modify#
# this script in order to make the solution work in your environment.#

# Substitute the name of your source and target machines, username, password, domain,      #
# and replication set name in the variable definitions below. (The replication set must    #
# already exist.)                                                                          #

$TheSource = "source_name";
$TheTarget = "target_name";
$User = "username";
$Pass = "password";
$TheDomain = "domain";
$TheRepSet = "repset_name";

login $TheSource $User $Pass $TheDomain;
login $TheTarget $User $Pass $TheDomain;
source $TheSource;

# The following commands determine the connection ID from the midnight connection and      #
# then disconnects it.                                                                     #
$MidConID = conid $TheRepSet to $TheTarget map base c:\midnight_mirror;
disconnect $MidConID;

# The following line connects the replication set and replicates the files to              #
# c:\noon_mirror\source_volume_name                                                        #
$NoonConID = connect $TheRepSet to $TheTarget map base c:\noon_mirror, nomirror;

# The following line starts a block checksum difference mirror.                           #
mirror start $NoonConID different, checksum;
```

SAMPLE_Midnight.dtcl

```
# ***Sample*** script to be run at midnight that stops one connection and starts          #
# a second connection.                                                                    #

# This sample batch file is provided as an example only. Because no two#
# environments or configurations are exactly the same, you MUST modify#
# this script in order to make the solution work in your environment.#

# Substitute the name of your source and target machines, username, password, domain,     #
# and replication set name in the variable definitions below. (The replication set must   #
# already exist.)                                                                         #

$TheSource = "source_name";
$TheTarget = "target_name";
$User = "username";
$Pass = "password";
$TheDomain = "domain";
$TheRepSet = "repset_name";

login $TheSource $User $Pass $TheDomain;
login $TheTarget $User $Pass $TheDomain;
source $TheSource;

# The following commands determine the connection ID from the noon connection and         #
# then disconnects it.                                                                    #
$NoonConID = conid $TheRepSet to $TheTarget map base c:\noon_mirror;
disconnect $NoonConID;

# The following line connects the replication set and replicates the files to             #
# c:\midnight_mirror\source_volume_name                                                   #
$MidnightConID = connect $TheRepSet to $TheTarget map base c:\midnight_mirror nomirror;

# The following line starts a block checksum difference mirror.                          #
mirror start $MidnightConID different, checksum;
```

These two scripts would have to be run at noon and midnight, respectively.



1a. At noon, the noon.dtcl script disconnects the midnight connection and starts a difference mirror and replication to the noon directory.
1b. At the same time, the tape backup process begins from the midnight directory.
2a. At midnight, the midnight.dtcl script disconnects the noon connection and starts a difference mirror and replication to the midnight directory.
2b. At the same time, the tape backup process begins from the noon directory.

You could expand the automation even further by using the Windows scheduler service to run the scripts. Or you could use a combination of Windows AT commands and Storage Mirroring DTCL commands to actually connect and disconnect the replication sets at different times. These methods would also allow multiple replicas to be stored on the different targets or on the same target server but in different directories.